

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号

特許第3062382号

(P3062382)

(45)発行日 平成12年7月10日(2000.7.10)

(24)登録日 平成12年4月28日(2000.4.28)

(51)IntCl.

G 0 6 K 9/20

識別記号

3 4 0

FI

G 0 6 K 9/20

3 4 0 L

請求項の数8(全8頁)

(21)出願番号 特願平5-309142

(22)出願日 平成5年12月9日(1993.12.9)

(65)公開番号 特開平7-160812

(43)公開日 平成7年6月23日(1995.6.23)

審査請求日 平成10年6月23日(1998.6.23)

(73)特許権者 000001007

キヤノン株式会社

東京都大田区下丸子3丁目30番2号

(72)発明者 原田 隆史

東京都大田区下丸子3丁目30番2号 キ

ヤノン株式会社内

(72)発明者 森 重樹

東京都大田区下丸子3丁目30番2号 キ

ヤノン株式会社内

(72)発明者 松林 一弘

東京都大田区下丸子3丁目30番2号 キ

ヤノン株式会社内

(74)代理人 100076428

弁理士 大塚 康徳 (外1名)

審査官 小林 勝広

最終頁に続く

(54)【発明の名称】 画像処理装置及び方法

(57)【特許請求の範囲】

【請求項1】 文字と線画が混在した画像を入力し、文字部分を抽出する画像処理装置であって、入力した画像を構成する有意なドットの集合体部分を膨張させる第1の膨張手段と、該第1の膨張手段を複数回実行し、膨張処理で加えられたドットの数、その前の膨張処理で加えられたドットの数と比較して減少するまで、前記第1の膨張手段による膨張処理を繰り返し行わせる第1の膨張回数制御手段と、前記第1の膨張回数制御手段により制御された回数より少なくとも2回少ない膨張処理を行った後のドットの集合体部分を取得する取得手段と、ドットの集合体部分を収縮させる収縮手段と、前記取得手段で取得したときにおける膨張回数よりも少

なくとも1回多い回数で、且つ、収縮処理で減少したドット数が、前回までの減少ドット数と比較して小さくなるまで、前記収縮手段を繰り返し行わせる収縮回数制御手段と、

前記収縮回数制御手段の制御の下で収縮処理を行った後のドットの集合体部分に対し、前記収縮回数制御手段により制御された回数と前記第1の膨張回数制御手段により制御された回数との差の回数分、前記膨張処理手段を実行する第2の膨張回数制御手段と、

該第2の膨張回数制御手段に制御された回数の膨張処理を行った後のドットの集合体部分に基づいて、前記入力された画像の文字部分を抽出する抽出手段とを有することを特徴とする画像処理装置。

【請求項2】 前記抽出手段により抽出される文字部分は、前記前記第2の膨張回数制御手段に制御された回数

の膨張処理を行った後のドットの集合体部分に対応する前記入力された画像の部分であることを特徴とする請求項１に記載の画像処理装置。

【請求項３】 前記膨張処理及び収縮処理は、前記有意なドットの集合体部分に対して垂直方向で行うことを特徴とする請求項１又は２のいずれかに記載の画像処理装置。

【請求項４】 前記膨張処理及び収縮処理は、前記有意なドットの集合体部分に対して水平方向で行うことを特徴とする請求項１又は２のいずれかに記載の画像処理装置。

【請求項５】 文字と線画が混在した画像を入力し、文字部分を抽出する画像処理方法であって、入力した画像を構成する有意なドットの集合体部分を膨張させる第１の膨張工程と、該第１の膨張工程を複数回実行し、膨張処理で加えられたドットの数、その前の膨張処理で加えられたドットの数と比較して減少するまで、前記第１の膨張工程による膨張処理を繰り返し行わせる第１の膨張回数制御工程と、前記第１の膨張回数制御工程により制御された回数より少なくとも２回少ない膨張処理を行った後のドットの集合体部分を取得する取得工程と、ドットの集合体部分を収縮させる収縮工程と、前記取得工程で取得したときにおける膨張回数よりも少なくとも１回多い回数で、且つ、収縮処理で減少したドット数が、前回までの減少ドット数と比較して小さくなるまで、前記収縮工程を繰り返し行わせる収縮回数制御工程と、前記収縮回数制御工程の制御の下で収縮処理を行った後のドットの集合体部分に対し、前記収縮回数制御工程により制御された回数と前記第１の膨張回数制御工程により制御された回数との差の回数分、前記膨張処理工程を実行する第２の膨張回数制御工程と、該第２の膨張回数制御工程に制御された回数の膨張処理を行った後のドットの集合体部分に基づいて、前記入力された画像の文字部分を抽出する抽出工程とを有することを特徴とする画像処理方法。

【請求項６】 前記抽出工程により抽出される文字部分は、前記前記第２の膨張回数制御工程に制御された回数の膨張処理を行った後のドットの集合体部分に対応する前記入力された画像の部分であることを特徴とする請求項５に記載の画像処理方法。

【請求項７】 前記膨張処理及び収縮処理は、前記有意なドットの集合体部分に対して垂直方向で行うことを特徴とする請求項５又は６のいずれか１項に記載の画像処理方法。

【請求項８】 前記膨張処理及び収縮処理は、前記有意なドットの集合体部分に対して水平方向で行うことを特徴とする請求項５又は６のいずれかに記載の画像処理方

法。

【発明の詳細な説明】

【０００１】

【発明の詳細な説明】 本願発明は画像処理装置及び方法、詳しくは画像中の線画と文字を分離する画像処理装置及び方法に関するものである。

【０００２】

【従来の技術】 地図や図面等、文字と線画が描かれている原稿をスキャナ等で読み取り、コンピュータ上で編集、文字検索等の処理する場合、これまでは、まず２値化、ラベリング、細線化等を行ない、ラベルの付けられた画素群に対し、画素数や長さをカウントして画素数の多いもの（長さの長いもの）を線画、短いものを文字を構成する要素として認識することにより、文字と線画を同時に抽出していた。

【０００３】

【発明が解決しようとする課題】 しかし、上述した従来の方法では文字の抽出と共に線画の抽出、および認識を行なっていたため、前処理に時間がかかりすぎるという欠点があった。

【０００４】

【課題を解決するための手段】 本発明は上記問題点に鑑みなされたものであり、線画と文字とが混在した画像を入力した場合、文字部分を効率良くそれから分離することを可能にした画像処理装置及び方法を提供しようとするものである。この課題を解決するため本発明の画像処理装置は以下に示す構成を備える。すなわち、文字と線画が混在した画像を入力し、文字部分を抽出する画像処理装置であって、入力した画像を構成する有意なドットの集合体部分を膨張させる第１の膨張手段と、該第１の膨張手段を複数回実行し、膨張処理で加えられたドットの数、その前の膨張処理で加えられたドットの数と比較して減少するまで、前記第１の膨張手段による膨張処理を繰り返し行わせる第１の膨張回数制御手段と、前記第１の膨張回数制御手段により制御された回数より少なくとも２回少ない膨張処理を行った後のドットの集合体部分を取得する取得手段と、ドットの集合体部分を収縮させる収縮手段と、前記取得手段で取得したときにおける膨張回数よりも少なくとも１回多い回数で、且つ、収縮処理で減少したドット数が、前回までの減少ドット数と比較して小さくなるまで、前記収縮手段を繰り返し行わせる収縮回数制御手段と、前記収縮回数制御手段の制御の下で収縮処理を行った後のドットの集合体部分に対し、前記収縮回数制御手段により制御された回数と前記第１の膨張回数制御手段により制御された回数との差の回数分、前記膨張処理手段を実行する第２の膨張回数制御手段と、該第２の膨張回数制御手段に制御された回数の膨張処理を行った後のドットの集合体部分に基づいて、前記入力された画像の文字部分を抽出する抽出手段とを有する。

【0005】また、本発明の画像処理方法は以下の工程を備える。すなわち、文字と線画が混在した画像を入力し、文字部分を抽出する画像処理方法であって、入力した画像を構成する有意なドットの集合体部分を膨張させる第1の膨張工程と、該第1の膨張工程を複数回実行し、膨張処理で加えられたドットの数が、その前の膨張処理で加えられたドットの数と比較して減少するまで、前記第1の膨張工程による膨張処理を繰り返し行わせる第1の膨張回数制御工程と、前記第1の膨張回数制御工程により制御された回数より少なくとも2回少ない膨張処理を行った後のドットの集合体部分を取得する取得工程と、ドットの集合体部分を収縮させる収縮工程と、前記取得工程で取得したときにおける膨張回数よりも少なくとも1回多い回数で、且つ、収縮処理で減少したドット数が、前回までの減少ドット数と比較して小さくなるまで、前記収縮工程を繰り返し行わせる収縮回数制御工程と、前記収縮回数制御工程の制御の下で収縮処理を行った後のドットの集合体部分に対し、前記収縮回数制御工程により制御された回数と前記第1の膨張回数制御工程により制御された回数との差の回数分、前記膨張処理工程を実行する第2の膨張回数制御工程と、該第2の膨張回数制御工程に制御された回数の膨張処理を行った後のドットの集合体部分に基づいて、前記入力された画像の文字部分を抽出する抽出工程とを有する。

【0006】かかる本発明の構成或いは工程において、入力した画像を構成する有意なドットの集合体部分を膨張させることを、各膨張処理で増加するドット数が前回と比較して減少するまで繰り返す。そして、この繰り返し回数よりも2回少ない回数で得られた膨張処理後のドットの集合体に対し、今度は収縮処理を繰り返す。この収縮処理の回数は、収縮処理を開始するときに得られた膨張処理の回数よりも1回多く、且つ、収縮処理で減少したドット数が、前回までの減少ドット数と比較して小

さくなるまで、前記収縮手段を繰り返し行わせる。そして、この収縮処理を行った後のドットの集合体部分に対し、これまでの収縮処理回数と膨張回数との差の回数分、膨張処理手段を実行する。この膨張処理を行った後のドットの集合体部分に基づいて、入力された画像の文字部分を抽出する。

【0007】

【実施例】以下、添付図面に従って本発明に係る実施例を詳細に説明する。図1に実施例の装置の構成を示し、図2に図1中のCPU3によって行なわれる処理手順（図1におけるプログラムメモリ6に格納されることになる）を示す。

【0008】図1において、1は画像を読み込むための画像入力装置であるスキャナ、2はスキャナ1からの画像データ等の各種データを一時保存するためのRAM、3は装置全体の制御を司るCPUで、プログラムメモリ6に記憶された制御プログラムに従って図2に示す処理を行う。4はRAM2中のデータにアクセスする際の位置を指定するアドレスバス、5はアドレスバスによって指定されたアドレスのデータの入出力や、画像入力装置1からのデータをRAM2あるいはCPU3に送るためのデータバス、7はVRAMであり、表示装置8で表示されるデータが置かれる。

【0009】図2を基に実施例の装置（CPU3）の動作内容を説明する。まずステップS1においてスキャナ1から画像データを読み込み、RAM2に保存する。次に、ステップS2において2値化を行う。ステップS3ではステップS2において出力された2値画像の任意の画素 $f(i, j)$ に対し以下の式（1）に従い膨張（拡散）処理を行う。

【0010】

【数1】

$$g_{ij} = \begin{cases} 1 : f_{ij}=1 \text{ 或いは} \\ \quad f_{ij} \text{ の周囲の 8 画素のいずれかが 1 のとき} \\ 0 : \text{その他} \end{cases} \quad \dots (1)$$

【0011】この処理により、線分の幅が太くなると同時に、文字列部分を塊にすることができる。図3にこの時の様子を示す。ここでは、線幅1、線間4、線の長さを75の場合を考える。このとき1回膨張処理を行うと1つの線幅は3（「2」増える）になり、線の部分の変化した画素数をカウントすると、468（＝ $\{75 \times 2 + (1+2) \times 2\} \times 3$ ）になる。

【0012】もう一度膨張処理を行うと、変化する画素数は480（＝ $\{(75+2) \times 2 + (3+2) \times 2\} \times 3 = 480$ ）となる。ただし、このとき、図3の下段に示す如く3本の線は一塊となってしまう。さらにもう一度膨張処理を行うと、 $\{(75+2+2) \times 2 + 15 \times 2\} = 188$ となり、変化する画素数の増加分が減るこ

とになる。

【0013】このことから、膨張処理は変化する画素数の増加が少なくなった時の回数より2回少ない回数行い、変化する画素数の増加が少なくなった時の2回前の画像をステップS3の出力とする。尚、線画が一本のみの場合には、上記のような膨張処理により変化する画素数の変化率が下がることがなくなるので、予め決められた回数だけ行う。

【0014】ステップS4ではステップS3を行った結果出力された画像の任意の画素 $f_{ij}$ に対し、以下の式（2）に従い収縮（縮退）処理を行う。

【0015】

【数2】

$$g_{ij} = \begin{cases} 0 : & f_{ij} = 0 \text{ 或いは} \\ & f_{ij} \text{の周囲の8画素のいずれかが0のとき} \\ 1 : & \text{その他} \end{cases}$$

( 2 )

【0016】この処理を行うことにより、線分の幅を補足することができる。図4にこの時の様子を示す。ステップS3の時と同様に变化した画素数をカウントして行くと、3回めには文字列部分が一塊になった部分だけが処理されるので、縮小処理により变化した画素数はそれまでと比較すると減少する。このことから、縮小処理は变化した画素数をカウントしていき、变化した画素数の減少が少なくなるまで繰り返される。

【0017】以上の結果、膨張、収縮処理の回数をそれぞれm、nとすると、 $m < n$ のとき線分が消去される。しかし、収縮処理を終了した結果の画像において文字列領域を示す画素の塊は、元画像における文字列領域よりも小さくなる。そこでステップS5において、再び膨張処理を $(n - m)$ 回行う。次に、ステップS5において出力された画像データから画素の塊を検出し、それを囲む矩形領域を検出する(ステップS6)。

【0018】次に、検出された領域に対応する元画像内の矩形領域を抽出し、その矩形領域から文字抽出を行う(ステップS7)。これは矩形領域に対し縦方向、横方向から沿って黒画素数を数えヒストグラムを生成し、ヒストグラムの谷の部分文字間の隙間の部分と判定し、山の部分を文字領域として判定することにより、文字抽出を行うことができる。

【0019】尚、上記実施例では、原稿画像をRAM2に格納した後に、2値化すると説明したが、スキャナ1からのデータをその時点で所定の閾値と比較するこ

とで2値化し、RAM2には予め2値化された画像を格納するようにしても良い。また、上記例に従えば、膨張処理、縮小処理を行い文字部分を残したが、かかる処理を行うと、RAM2には未加工の原画画像を保持しておくメモリ空間が必要になるか、再度原稿画像を読み取りを行う必要になる。そこで、メモリ容量の単価の低いハードディスク装置等に原画像を一旦記憶させておく。そして、上記処理でRAM2上で文字領域の特定する情報を抽出した後、先に待避しておいた画像データを再びRAM2に展開し、文字の切り出し及び認識するようにしても良い。これは、以下に説明する他の実施例でも同様である。

【0020】[第2の実施例の説明] 次に、図5のフローチャートを基に第2の実施例を説明する。尚、本第2の実施例を実現するためには、同図のフローチャートに基づくプログラムが図1のプログラムメモリ6に格納されている。まず、ステップS11において、画像入力装置1から画像データを読み込みRAM2に保存する。次に、ステップS12において2値化を行う。ステップS13ではステップS12において出力された2値画像の任意の画素 $f_{ij}$ に対し以下の式(3)に従い膨張(拡散)処理を行う(処理後の画素位置 $g_{ij}$ の2値を定めている)。

【0021】

【数3】

$$g_{ij} = \begin{cases} 1 : & f_{ij} = 1 \text{ 或いは} \\ & f_{ij} \text{の左右の画素のいずれかが1のとき} \\ 0 : & \text{その他} \end{cases} \quad \dots (3)$$

【0022】この処理により、横方向への膨張処理が行われ、縦線の幅が大きくなると同時に、横方向に並んだ文字列部分を塊にすることができる。膨張処理の回数は、第1実施例で示した方法により決定される。次にステップS14ではステップS13において出力された画像の

任意の画素 $f_{ij}$ に対し以下の式(4)に従い収縮処理を行う。

【0023】

【数4】

$$g_{ij} = \begin{cases} 0 : & f_{ij} = 0 \text{ 或いは} \\ & f_{ij} \text{の左右の画素のいずれかが0のとき} \\ 1 : & \text{その他} \end{cases} \quad (4)$$

【0024】収縮処理の回数は、第1実施例で示したものと同一方法により決定される。式(3)、式(4)による膨張、縮小処理の回数をそれぞれ $m_1$ 、 $n_1$ とすると、 $m_1 < n_1$ のとき縦線が消去される。収縮処理を終了した結果の画像において文字列領域を示す画素の塊は、元画像における文字列領域よりも小さくなる。そこでステップS15において、再び式(3)に従った膨張処理を $(n_1 - m_1)$ 回行う。

$$g_{ij} = \begin{cases} 1 : f_{ij}=1 \text{ 或いは} \\ \quad f_{ij} \text{の上下の画素のいずれかが1のとき} \\ 0 : \text{その他} \end{cases} \quad \dots (5)$$

【0027】この処理により、縦方向への膨張処理が行われた横線の幅が太くなると同時に、縦方向へ並んだ文字列部分を塊にすることができる。次に、ステップS17ではステップS16において出力された画像の任意の

【0025】次に、ステップS16ではステップS15において出力された画像の任意の画素 $f_{ij}$ に対し以下の式(5)に従い膨張処理を行う。なお、膨張処理回数についてはステップS13と同じ方法により決められる。

【0026】

【数5】

画素 $f_{ij}$ に対し以下の式(6)に従い収縮処理を行う。

【0028】

【数6】

$$g_{ij} = \begin{cases} 0 : f_{ij} = 0 \text{ 或いは} \\ \quad f_{ij} \text{の上下の画素のいずれかが0のとき} \\ 1 : \text{その他} \end{cases}$$

(6)

【0029】尚、収縮処理の回数についてはステップS14と同じ方法により決められる。式(5)、式(6)による膨張、収縮処理の回数をそれぞれ $m_2$ 、 $n_2$ とすると、 $m_2 < n_2$ のとき横線が消去される。収縮処理を終了した結果の画像において文字列領域を示す画素の塊は、元画像における文字列領域よりも小さくなる。そこでステップS18において、再び式(5)に従って膨張処理を $(n_2 - m_2)$ 回行う。次に、ステップS17において出力された画像データから画素の塊を検出し、それを囲む矩形領域を検出する(ステップS19)。

【0030】次に、検出された領域に対応する元画像内の矩形領域を抽出し、その矩形領域から文字抽出を行う(ステップS20)。ステップS20の文字抽出については、実施例1に述べた方法と同じ方法により行う。

【第3の実施例の説明】次に、図6のフローチャートを基に本発明の第3実施例を説明する。ただし、この処理内容に従ったプログラムがプログラムメモリ6に格納されているのは、上記第2の実施例と同様である。

【0031】さて、本第3の実施例では、上記第2の実施例で述べた方法により、ステップ21からステップ29で元画像から文字列領域を含む矩形領域を抽出する(それぞれは図5におけるステップS11からステップS19に対応する)。次にステップ130より、この矩形領域内で連結している画素に同一の数字を割り当てて行く(ラベリング)。次のステップ11では、ラベリングされた画素群の位置関係から文字の抽出を行う。

【0032】例えば、文字列イメージ“a j”に対しラ

ベリングを行うと“a”を構成する画素には1、“j”の上の点を構成する画素には2、その下のストロークを構成する画素に対しては3という番号が付けられる。次のステップ10において、同一ラベルの付けられた画素群を囲む矩形と画素群の重心を求める。そしてその結果、2と3の重心間の距離は1との重心間の距離より大分小さいので、2と3で1文字を構成し、1と2、3の間は広く空いているので1は1文字である、というように判定し、(ステップ11)、それぞれ文字として抽出される。

【0033】尚、元画像から文字列領域を含む矩形領域を抽出する方法は、第1の実施例のステップ1からステップ6を用いることも可能である。以上は特に黒に注目して2値化したのが、地図のように色のある文字が含まれる場合には、ある色に注目して上記の処理を行うことによりその色の文字を抽出することができる。

【0034】以上説明したように本実施例よれば、線分を消去し、文字の含まれる領域を抽出し、そのあとで文字の抽出を行うことにより、従来より処理時間の短縮を行うことができる。尚、実施例では単独の装置として説明したが、画像読み取り部と上記処理を行う処理部とを独立させたシステム、例えばイメージスキャナ装置とパーソナルコンピュータで構成されるシステムにも適用できることは言うまでもない。従って、本願発明は装置或はシステムに外部からプログラムを供給することで実現する場合にも適用可能であることは当業者であれば容易に推察されよう。

【0035】尚、実施例では線画が1ドットの幅を有するとして説明したが、これは読み取り解像度や原稿中の線分の太さで変化するので、操作者が適宜指定するようにしても良い。このとき、その指定内容に応じて上記処理における収縮処理の回数等が変化するのは、容易に推察されよう。さらには、実施例ではスキャナを介して原稿画像を直接入力したが、例えばフロッピーディスクなどに記憶された文字と線画の混在画像でも良く、場合によっては回線を介して受信するようにしても良い。

#### 【0036】

【発明の効果】以上説明したように本発明によれば、線画と文字とが混在した画像を入力した場合、文字部分を効率良くそれらを分離することが可能になる。

#### 【図面の簡単な説明】

【図1】実施例における装置のブロック構成図である。

【図2】第1の実施例における処理内容を示すフローチャートである。

ャートである。

【図3】第1の実施例の膨張処理における画像の推移を示す図である。

【図4】第1の実施例の収縮処理における画像の推移を示す図である。

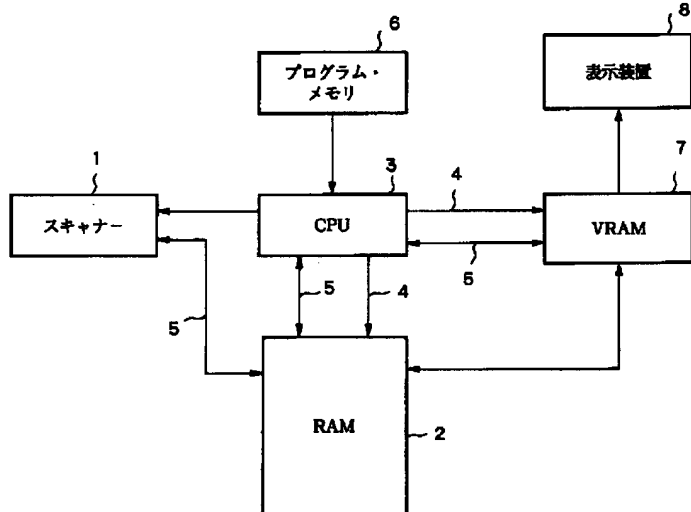
【図5】第2の実施例における処理内容を示すフローチャートである。

【図6】第3の実施例における処理内容を示すフローチャートである。

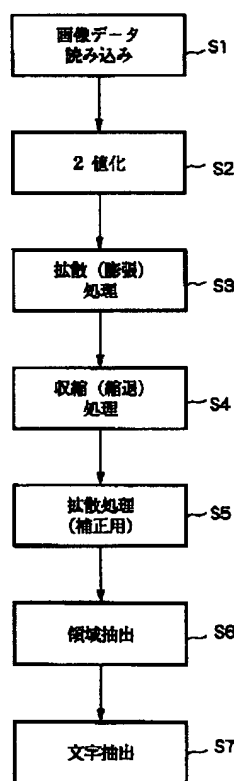
#### 【符号の説明】

- 1 スキャナ
- 2 RAM
- 3 CPU
- 4 VRAM
- 5 表示装置
- 6 プログラムメモリ

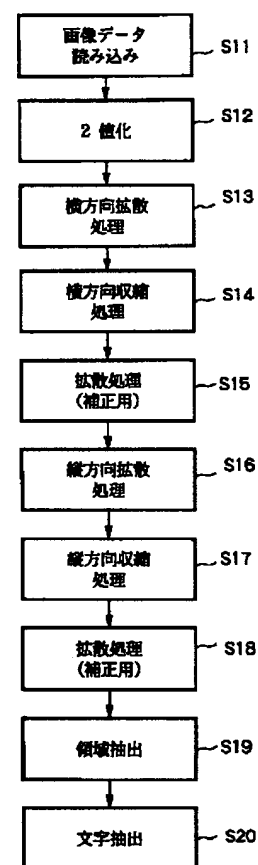
【図1】



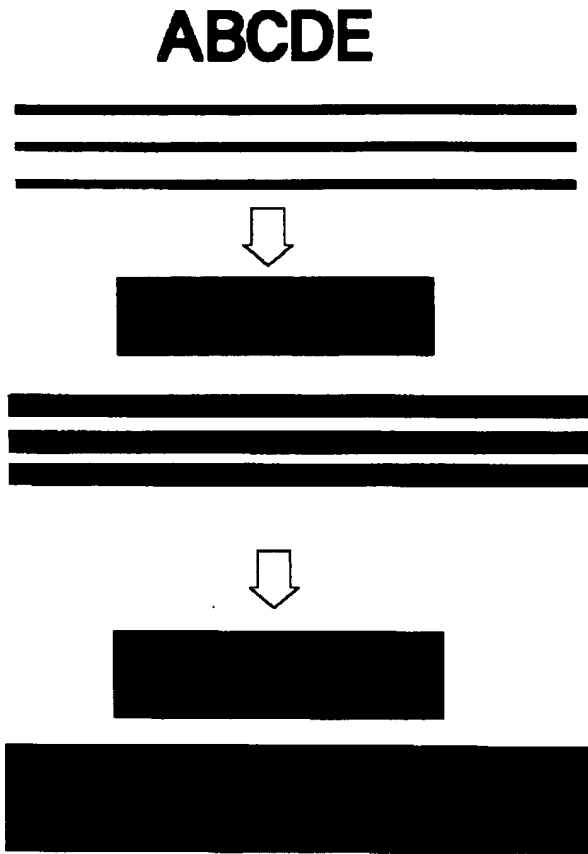
【図2】



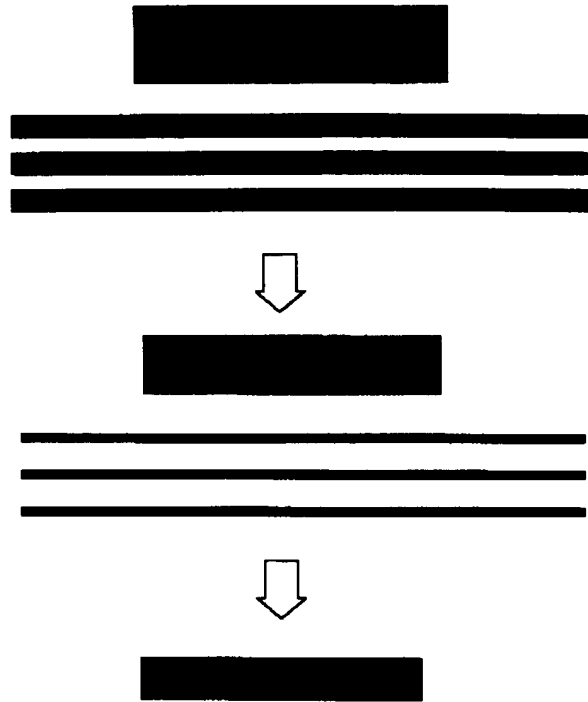
【図5】



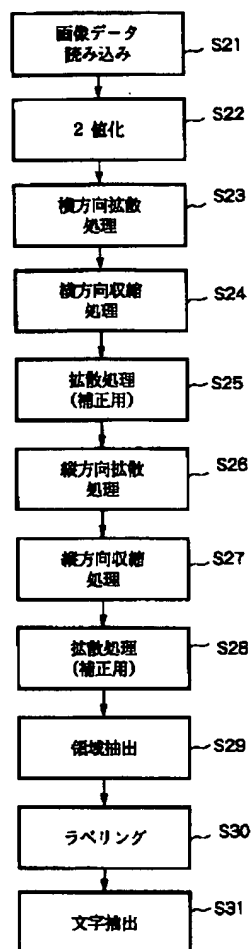
【图3】



【图4】



【図 6】



フロントページの続き

(72)発明者 異 栄作  
東京都大田区下丸子 3 丁目 30 番 2 号 キ  
ヤノン株式会社内

(72)発明者 島田 和俊  
東京都大田区下丸子 3 丁目 30 番 2 号 キ  
ヤノン株式会社内

(72)発明者 砂川 伸一  
東京都大田区下丸子 3 丁目 30 番 2 号 キ  
ヤノン株式会社内

(72)発明者 長崎 克彦  
東京都大田区下丸子 3 丁目 30 番 2 号 キ  
ヤノン株式会社内

(56)参考文献 特開 平 3 - 260787 ( J P , A )  
特開 平 1 - 266682 ( J P , A )  
特開 平 5 - 143774 ( J P , A )

(58)調査した分野(Int.Cl.<sup>7</sup>, D B 名)  
G06K 9/18 - 9/44